Geometric Model of Helical Auxetic Capacitor Sensors

The helical auxetic capacitive sensor is composed of one elastic fibre and one inextensible copper wire. At the initial state (ε_x=0), the elastic fibre is straight and the inextensible fibre is coiled around it.

Input parameters:

```
> wire_diameter := 0.2;
  initial_elastic_diameter := 2;
  dielectric_thickness := 0.05;
  initial_pitch := 4;
  elastic_poissons_ratio := 0.4;
  final_strain := 0.9;
  number_of_turns := 2;
  run_name := "E3R1-04";
  with(plots) :
  with(ColorTools) :
  plotsetup(default) :
```

$$wire\_diameter := 0.2$$

$$initial\_elastic\_diameter := 2$$

$$dielectric\_thickness := 0.05$$

$$initial\_pitch := 4$$

$$elastic\_poissons\_ratio := 0.4$$

$$final\_strain := 0.9$$

$$number\_of\_turns := 2$$

$$run\_name := \text{"E3R1-04"} \tag{1}$$

First, define the equation for a helix in Cartesian (x, y, z) coordinates, parameterized by (t, u). The parameter t ranges from 0 to 2πn, where n is the number of total turns. u ranges from 0 to 2π. Plot an example of a single-turn helix with major radius r_M, minor radius r_m, and pitch p.

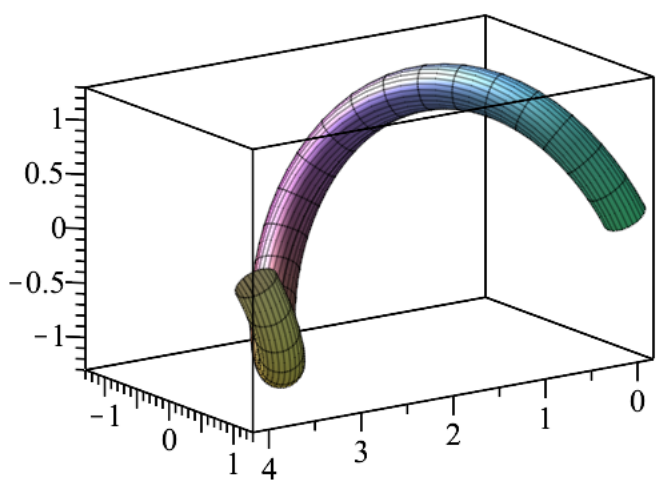
$$> helix := (t, u, \theta, r_M, r_m, p) \rightarrow \left[\frac{1}{2 \cdot \pi} p \cdot t + \frac{r_M \cdot r_m \cdot \sin(u)}{\mathrm{sqrt}\left(r_M^{\,2} + \left(\frac{p}{2 \cdot \pi}\right)^2\right)}, \right.$$

$$r_M \cdot \cos(t+\theta) - r_m \cdot \cos(t+\theta) \cdot \cos(u) + \frac{p \cdot r_m \cdot \sin(t+\theta) \cdot \sin(u)}{2 \cdot \pi \cdot \mathrm{sqrt}\left(r_M^{\,2} + \left(\frac{p}{2 \cdot \pi}\right)^2\right)},$$

$$\left. r_M \cdot \sin(t+\theta) - r_m \cdot \sin(t+\theta) \cdot \cos(u) - \frac{p \cdot r_m \cdot \cos(t+\theta) \cdot \sin(u)}{2 \cdot \pi \cdot \mathrm{sqrt}\left(r_M^{\,2} + \left(\frac{p}{2 \cdot \pi}\right)^2\right)} \right];$$

$$plot3d\left(helix\left(t, u, 0, \frac{(initial\_elastic\_diameter + wire\_diameter)}{2}, wire\_diameter, initial\_pitch\right), t = 0 ..2\cdot\pi, u = 0 ..2\cdot\pi, scaling = constrained\right)$$

$$helix := \left(t, u, \theta, r_M, r_m, p\right) \mapsto \left[\frac{p\cdot t}{2\cdot\pi} + \frac{r_M\cdot r_m\cdot\sin(u)}{\sqrt{r_M^{\ 2} + \frac{p^2}{4\cdot\pi^2}}}, r_M\cdot\cos(t+\theta) - r_m\cdot\cos(t+\theta)\cdot\cos(u) + \frac{p\cdot r_m\cdot\sin(t+\theta)\cdot\sin(u)}{2\cdot\pi\cdot\sqrt{r_M^{\ 2} + \frac{p^2}{4\cdot\pi^2}}}, r_M\cdot\sin(t+\theta) - r_m\cdot\sin(t+\theta)\cdot\cos(u) - \frac{p\cdot r_m\cdot\cos(t+\theta)\cdot\sin(u)}{2\cdot\pi\cdot\sqrt{r_M^{\ 2} + \frac{p^2}{4\cdot\pi^2}}}\right]$$



We solve for the equations of the helix sliced by a plane in the *x* axis, which is will be needed later to generate the cross-section images. This is done by solving the helix x equation for t, and substituting it into the other helix equations to eliminate t. Now, new parameter x defines the location of the plane, and the equations trace out a helical cross-section.

> $t\_cross\_section := solve\left(x = helix\left(t, u, \theta, r_M, r_m, p\right)[1], t\right);$
> $helix\_cross\_section := unapply\left(simplify\left(subs\left(\{t = t\_cross\_section\}, helix\left(t, u, \theta, r_M, r_m, p\right)\right)\right), x, u, \theta, r_M, r_m, p\right):$

$$t\_cross\_section := -\frac{2\left(2\, r_M r_m \sin(u)\, \pi - x\sqrt{4\,\pi^2 r_M^{\,2} + p^2}\right)\pi}{p\sqrt{4\,\pi^2 r_M^{\,2} + p^2}} \tag{2}$$

We need some helper functions. First, the function for the length of a helical path with radius r, ptich p, and n turns.

> $helical\_length := (r, p, n) \rightarrow n \cdot \mathrm{sqrt}\left((2\cdot\pi\cdot r)^2 + p^2\right)$

$$helical\_length := (r, p, n) \mapsto n\cdot\sqrt{4\cdot\pi^2\cdot r^2 + p^2} \tag{3}$$

The centre length of the wire is equal to the helical length, defined by the diameters of the two elements, pitch, and number of turns.

> $l_{wire} := helical\_length\left(\frac{1}{2}\cdot\left(d_{elastic,\, i} + d_{wire}\right), p_i, n_{turn}\right)$

$$l_{wire} := n_{turn}\sqrt{4\,\pi^2\left(\frac{d_{elastic,i}}{2} + \frac{d_{wire}}{2}\right)^2 + p_i^{\,2}} \tag{4}$$

Next, we wish to obtain an equation for the helical major radius of the wire as a function of sensor strain ε_x. We assume this is irrespective of the elastic element. We equate the helical length of the initial wire coil with the helical length of the wire coil strained to ε_x. Then, we can solve for the strained major raidus $r_{M,wire}$.

> $r_{M,\, wire} := solve\left(l_{wire} = helical\_length\left(r, p, n_{turn}\right), r\right)[1]:$
> $r_{M,wire} := unapply\left(simplify\left(subs\left(p = p_i\cdot\left(1 + \varepsilon_x\right), r_{M,wire}\right)\right), \varepsilon_x\right);$

$$r_{M,wire} := \varepsilon_x \mapsto \frac{\sqrt{\left(d_{elastic,i} + d_{wire}\right)^2\cdot\pi^2 - p_i^{\,2}\cdot\varepsilon_x\cdot\left(\varepsilon_x + 2\right)}}{2\cdot\pi} \tag{5}$$

From the helical major radius of the wire we may calculate the helical major radius of the elastic.

> $r_{M,\, elastic} := \left(\varepsilon_x\right) \rightarrow \frac{1}{2}\left(d_{elastic} + d_{wire}\right) - r_{M,\, wire}\left(\varepsilon_x\right)$

$$r_{M,elastic} := \varepsilon_x \mapsto \frac{d_{elastic}}{2} + \frac{d_{wire}}{2} - r_{M,wire}\left(\varepsilon_x\right) \tag{6}$$

Some helper functions to calculate strain and raidal strain of an isotropic rod from axial strain and its Poisson's ratio.

> $strain := (l, l_0) \rightarrow \frac{(l-l_0)}{l_0};$

$poisson\_strain\_y := (\varepsilon_x, \nu) \rightarrow -\left(1 - (1+\varepsilon_x)^{-\nu}\right);$

$$strain := (l, l_0) \mapsto \frac{l - l_0}{l_0}$$

$$poisson\_strain\_y := (\varepsilon_x, \nu) \mapsto -1 + (\varepsilon_x + 1)^{-\nu} \tag{7}$$

The following strain expressions are for the elastic element in transverse and axial directions.

> $elastic\_radial\_strain := strain(d_{elastic}, d_{elastic,\,i});$

$elastic\_axial\_strain := strain(l_{elastic}, l_{elastic,\,i})$

$$elastic\_radial\_strain := \frac{d_{elastic} - d_{elastic,i}}{d_{elastic,i}}$$

$$elastic\_axial\_strain := \frac{l_{elastic} - l_{elastic,i}}{l_{elastic,i}} \tag{8}$$

We equate the above expressions using the Poisson's ratio function defined in **(7)**.

> $elastic\_strain := elastic\_radial\_strain = poisson\_strain\_y(elastic\_axial\_strain, \nu_{elastic}) :$

$elastic\_diameter := solve(elastic\_strain, d_{elastic})$

$$elastic\_diameter := \left(\frac{l_{elastic} - l_{elastic,i}}{l_{elastic,i}} + 1\right)^{-\nu_{elastic}} d_{elastic,i} \tag{9}$$

We make expressions for the elastic diameter, from only linear extension. This is compared to the major diameter of the wire coil to check if there should be a gap between the wire coil and elastic.

> $elastic\_diameter\_from\_extension := simplify(subs(\{l_{elastic} = l_{elastic,i} \cdot (1+\varepsilon_x)\},$
> $elastic\_diameter));$

$elastic\_major\_radius := subs(\{d_{elastic} = elastic\_diameter\_from\_extension\}, r_{M,elastic}(\varepsilon_x)) ;$

$$elastic\_diameter\_from\_extension := (1+\varepsilon_x)^{-\nu_{elastic}} d_{elastic,i}$$

$$elastic\_major\_radius := \frac{(1+\varepsilon_x)^{-\nu_{elastic}} d_{elastic,i}}{2} + \frac{d_{wire}}{2} - \frac{\sqrt{(d_{elastic,i} + d_{wire})^2 \pi^2 - p_i^2 \varepsilon_x (\varepsilon_x + 2)}}{2\pi} \tag{10}$$

The elastic length from its helical path is a function of the elastic diameter.

> $pitch\_from\_strain := (\varepsilon) \rightarrow p_i \cdot (1+\varepsilon);$

$elastic\_length\_from\_helical\_path := helical\_length\left(r_{M,\,elastic}(\varepsilon_x), pitch\_from\_strain(\varepsilon_x), n_{turn}\right)$

$$pitch\_from\_strain := \varepsilon \mapsto p_i \cdot (\varepsilon + 1)$$

$$elastic\_length\_from\_helical\_path := n_{turn} \sqrt{4\,\pi^2 \left(\frac{d_{elastic}}{2} + \frac{d_{wire}}{2} - \frac{\sqrt{\left(d_{elastic,i} + d_{wire}\right)^2 \pi^2 - p_i^2\, \varepsilon_x\, (\varepsilon_x + 2)}}{2\,\pi}\right)^2 + p_i^2\, (1 + \varepsilon_x)^2} \tag{11}$$

The elastic length is also defined from Poisson's ratio in **(9)**, which is also dependent on the elastic diameter.

> $elastic\_length\_from\_poisson := solve\left(d_{elastic} = elastic\_diameter, l_{elastic}\right)$

$$elastic\_length\_from\_poisson := \mathrm{e}^{-\frac{\ln\left(\frac{d_{elastic}}{d_{elastic,i}}\right)}{\nu_{elastic}}}\; l_{elastic,i} \tag{12}$$

We need to solve for the elastic diameter, by equating **(11)** and **(12)**.

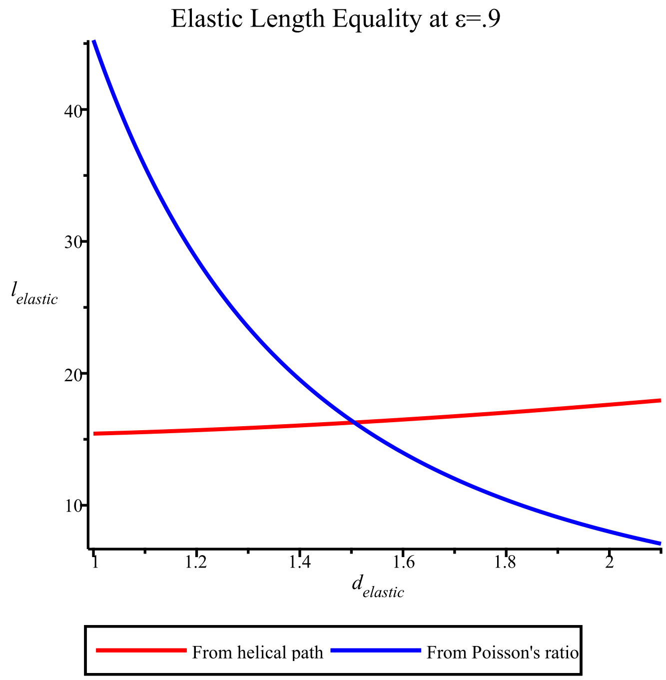
> $elastic\_length\_equality := elastic\_length\_from\_helical\_path = elastic\_length\_from\_poisson :$
> $elastic\_length\_equality := subs\left(l_{elastic,i} = p_i \cdot n_{turn}, elastic\_length\_equality\right)$

$$elastic\_length\_equality := n_{turn} \sqrt{4\,\pi^2 \left(\frac{d_{elastic}}{2} + \frac{d_{wire}}{2} - \frac{\sqrt{\left(d_{elastic,i} + d_{wire}\right)^2 \pi^2 - p_i^2\, \varepsilon_x\, (\varepsilon_x + 2)}}{2\,\pi}\right)^2 + p_i^2\, (1 + \varepsilon_x)^2} = \mathrm{e}^{-\frac{\ln\left(\frac{d_{elastic}}{d_{elastic,i}}\right)}{\nu_{elastic}}}\; p_i\, n_{turn} \tag{13}$$

This is equivalent to finding the point where the elastic lengths from both equations are equal. The corresponding elastic diameter is used. A graphical interpretation is shown:

> $with(plots) :$
> $p1 := plot\left(subs\left(\{d_{wire} = wire\_diameter, d_{elastic,i} = initial\_elastic\_diameter, p_i = initial\_pitch, \varepsilon_x = final\_strain, n_{turn} = number\_of\_turns\}, elastic\_length\_from\_helical\_path\right), d_{elastic} = 1 ..2.1, color = red\right) :$
> $p2 := plot\left(subs\left(\{d_{elastic,i} = initial\_elastic\_diameter, \nu_{elastic} = elastic\_poissons\_ratio, l_{elastic,i} = number\_of\_turns \cdot initial\_pitch\}, elastic\_length\_from\_poisson\right), d_{elastic} = 1 ..2.1, color = blue\right) :$
> $display(p1, p2, labels = [d_{elastic}, l_{elastic}], legend =$ ["From helical path", "From Poisson's ratio"],

$title = cat(\text{"Elastic Length Equality at ε=", } convert(final\_strain, string)))$



> $elastic\_diameter\_from\_strain := (\varepsilon) \rightarrow fsolve\Big(subs\Big(\{d_{wire} = wire\_diameter, d_{elastic,\ i} = initial\_elastic\_diameter, p_i = initial\_pitch, n_{turn} = 1, \nu_{elastic} = elastic\_poissons\_ratio, \varepsilon_x = \varepsilon\}, elastic\_length\_equality\Big), d_{elastic}, 0\,..initial\_elastic\_diameter\Big) :$

Functions for Plotting

The following two functions generate the equations for the helical sensor geometry, with and without forced contact.

> $SensorGeometryForcedContact := \mathbf{proc}(d_e, d_w, p, \nu, n, \varepsilon)$

local $elastic\_diameter\_from\_strain, helix\_wire\_outer, helix\_wire\_inner, helix\_elastic, helices, max\_length, max\_strain, strain\_clipped;$

$max\_length := subs(\{n_{turn} = n, d_{elastic,i} = d_e, d_{wire} = d_w, p_i = p\}, l_{wire});$

$$max\_strain := \frac{(max\_length \quad n \cdot p)}{n \cdot p};$$

if $\varepsilon > (max\_strain \cdot 0.9999)$ **then** # *detect if strain is too large, and clip*

$strain\_clipped := max\_strain \cdot 0.9999;$

else

$strain\_clipped := \varepsilon;$

end if;

$elastic\_diameter\_from\_strain := (\varepsilon) \rightarrow fsolve\big(subs\big(\{d_{wire} = d_w, d_{elastic,i} = d_e, p_i = p, n_{turn} = n, \nu_{elastic} = \nu, \varepsilon_x = \varepsilon\}, elastic\_length\_equality\big), d_{elastic}, 0\,..d_e\big);$

$$helix\_wire\_outer := helix\left(t, u, 0, r_{M,wire}(\varepsilon_x), \frac{d_{wire}}{2}, pitch\_from\_strain(\varepsilon_x)\right);$$

$helix\_wire\_outer := subs(\{d_{elastic} = elastic\_diameter\_from\_strain(strain\_clipped)\}, helix\_wire\_outer);$

```
helix_wire_outer := subs({ε_x = strain_clipped}, helix_wire_outer) ;
helix_wire_outer := subs({d_wire = d_w, p_i = p, d_elastic,i = d_e}, helix_wire_outer);
helix_elastic := helix(t, u, π, r_M,elastic(ε_x), d_elastic/2, pitch_from_strain(ε_x));
helix_elastic := subs({d_elastic = elastic_diameter_from_strain(strain_clipped)}, helix_elastic);
helix_elastic := subs({ε_x = strain_clipped}, helix_elastic);
helix_elastic := subs({d_wire = d_w, p_i = p, d_elastic,i = d_e}, helix_elastic);
helices := [helix_wire_outer, helix_elastic];
end proc:
SensorGeometry := proc(d_e, d_w, p, ν, n, ε)
local elastic_diameter_from_strain, helix_wire_outer, helix_wire_inner, helix_elastic, helices,
    elastic_diameter_from_extension, elastic_major_radius, max_length, max_strain,
    strain_clipped;
max_length := subs({n_turn = n, d_elastic,i = d_e, d_wire = d_w, p_i = p}, l_wire);
max_strain := (max_length − n·p) / (n·p);
if ε > (max_strain·0.9999) then # detect if strain is too large, and clip
strain_clipped := max_strain·0.9999;
else
strain_clipped := ε;
end if;
elastic_diameter_from_strain := (ε) → fsolve(subs({d_wire = d_w, d_elastic,i = d_e, p_i = p, n_turn = n,
    ν_elastic = ν, ε_x = ε}, elastic_length_equality), d_elastic, 0 ..d_e);
helix_wire_outer := helix(t, u, 0, r_M,wire(ε_x), d_wire/2, pitch_from_strain(ε_x));
helix_wire_outer := subs({d_elastic = elastic_diameter_from_strain(ε)}, helix_wire_outer);
helix_wire_outer := subs({ε_x = strain_clipped}, helix_wire_outer) ;
helix_wire_outer := subs({d_wire = d_w, p_i = p, d_elastic,i = d_e}, helix_wire_outer);
elastic_diameter_from_extension := simplify(subs({l_elastic = l_elastic,i·(1 + ε_x)},
    elastic_diameter));
elastic_diameter_from_extension := subs({ε_x = strain_clipped, ν_elastic = ν, d_elastic,i = d_e},
    elastic_diameter_from_extension);
elastic_major_radius := subs({d_elastic = elastic_diameter_from_extension}, r_M,elastic(ε_x)) ;
elastic_major_radius := subs({d_elastic,i = d_e, d_wire = d_w, p_i = p, ν_elastic = ν, ε_x = strain_clipped},
    elastic_major_radius);
if elastic_major_radius < 0 then
helix_elastic := helix(t, u, π, 0, elastic_diameter_from_extension/2,
```

```
    pitch_from_strain(strain_clipped) );
  else
  helix_elastic := helix( t, u, π, r_{M,elastic}(ε_x), d_elastic/2, pitch_from_strain(ε_x) );
  helix_elastic := subs( {d_elastic = elastic_diameter_from_strain(strain_clipped)}, helix_elastic );
  end if;
  helix_elastic := subs( {ε_x = strain_clipped}, helix_elastic );
  helix_elastic := subs( {d_wire = d_w, p_i = p, d_{elastic,i} = d_e}, helix_elastic );
  helices := [helix_wire_outer, helix_elastic];
  end proc:
```

The following two functions generate the equations for the cross-section of the sensor, with and without forced contact.

```
> SensorCrossSectionForcedContact := proc(x, d_e, d_w, t_i, p, ν, n, ε)
  local elastic_diameter_from_strain, cs_wire_outer, cs_wire_inner, cs_elastic, cross_sections,
    max_length, max_strain, strain_clipped;
  max_length := subs( {n_turn = n, d_{elastic,i} = d_e, d_wire = d_w, p_i = p}, l_wire );
  max_strain := (max_length − n·p) / (n·p);
  if ε > (max_strain·0.9999) then # detect if strain is too large, and clip
  strain_clipped := max_strain·0.9999;
  else
  strain_clipped := ε;
  end if;
  elastic_diameter_from_strain := (ε) → fsolve( subs( {d_wire = d_w, d_{elastic,i} = d_e, p_i = p, n_turn = n,
    ν_elastic = ν, ε_x = ε}, elastic_length_equality ), d_elastic, 0 ..d_e );
  cs_wire_outer := helix_cross_section( x, u, 0, r_{M, wire}(ε_x), d_wire/2, pitch_from_strain(ε_x) );
  cs_wire_outer := subs( {d_elastic = elastic_diameter_from_strain(strain_clipped)},
    cs_wire_outer );
  cs_wire_outer := subs( {ε_x = strain_clipped}, cs_wire_outer ) ;
  cs_wire_outer := subs( {d_wire = d_w, p_i = p, d_{elastic,i} = d_e}, cs_wire_outer );
  cs_wire_inner := helix_cross_section( x, u, 0, r_{M,wire}(ε_x), d_wire/2 − t_i, pitch_from_strain(ε_x) );
  cs_wire_inner := subs( {d_elastic = elastic_diameter_from_strain(strain_clipped)},
    cs_wire_inner );
  cs_wire_inner := subs( {ε_x = strain_clipped}, cs_wire_inner ) ;
  cs_wire_inner := subs( {d_wire = d_w, p_i = p, d_{elastic,i} = d_e}, cs_wire_inner );
  cs_elastic := helix_cross_section( x, u, π, r_{M, elastic}(ε_x), d_elastic/2, pitch_from_strain(ε_x) );
```

$cs\_elastic := subs\left(\left\{d_{elastic} = elastic\_diameter\_from\_strain(strain\_clipped)\right\}, cs\_elastic\right);$
$cs\_elastic := subs\left(\left\{\varepsilon_x = strain\_clipped\right\}, cs\_elastic\right);$
$cs\_elastic := subs\left(\left\{d_{wire} = d_w, p_i = p, d_{elastic,i} = d_e\right\}, cs\_elastic\right);$
$cross\_sections := [cs\_wire\_outer, cs\_wire\_inner, cs\_elastic];$
end proc: $SensorCrossSection :=$ **proc**$\left(x, d_e, d_w, t_i, p, \nu, n, \varepsilon\right)$
local $elastic\_diameter\_from\_strain, cs\_wire\_outer, cs\_wire\_inner, cs\_elastic, cross\_sections,$
$max\_length, max\_strain, strain\_clipped, elastic\_diameter\_from\_extension,$
$elastic\_major\_radius;$
$max\_length := subs\left(\left\{n_{turn} = n, d_{elastic,i} = d_e, d_{wire} = d_w, p_i = p\right\}, l_{wire}\right);$
$max\_strain := \dfrac{(max\_length - n \cdot p)}{n \cdot p};$
if $\varepsilon > (max\_strain \cdot 0.9999)$ **then** *# detect if strain is too large, and clip*
$strain\_clipped := max\_strain \cdot 0.9999;$
else
$strain\_clipped := \varepsilon;$
end if;
$elastic\_diameter\_from\_strain := (\varepsilon) \rightarrow fsolve\left(subs\left(\left\{d_{wire} = d_w, d_{elastic,i} = d_e, p_i = p, n_{turn} = n,\right.\right.\right.$
$\left.\left.\left.\nu_{elastic} = \nu, \varepsilon_x = \varepsilon\right\}, elastic\_length\_equality\right), d_{elastic}, 0 .. d_e\right);$
$cs\_wire\_outer := helix\_cross\_section\left(x, u, 0, r_{M,wire}\left(\varepsilon_x\right), \dfrac{d_{wire}}{2}, pitch\_from\_strain\left(\varepsilon_x\right)\right);$
$cs\_wire\_outer := subs\left(\left\{d_{elastic} = elastic\_diameter\_from\_strain(strain\_clipped)\right\},\right.$
$\left.cs\_wire\_outer\right);$
$cs\_wire\_outer := subs\left(\left\{\varepsilon_x = strain\_clipped\right\}, cs\_wire\_outer\right);$
$cs\_wire\_outer := subs\left(\left\{d_{wire} = d_w, p_i = p, d_{elastic,i} = d_e\right\}, cs\_wire\_outer\right);$
$cs\_wire\_inner := helix\_cross\_section\left(x, u, 0, r_{M,wire}\left(\varepsilon_x\right), \dfrac{d_{wire}}{2} - t_i, pitch\_from\_strain\left(\varepsilon_x\right)\right);$
$cs\_wire\_inner := subs\left(\left\{d_{elastic} = elastic\_diameter\_from\_strain(strain\_clipped)\right\},\right.$
$\left.cs\_wire\_inner\right);$
$cs\_wire\_inner := subs\left(\left\{\varepsilon_x = strain\_clipped\right\}, cs\_wire\_inner\right);$
$cs\_wire\_inner := subs\left(\left\{d_{wire} = d_w, p_i = p, d_{elastic,i} = d_e\right\}, cs\_wire\_inner\right);$
$elastic\_diameter\_from\_extension := simplify\left(subs\left(\left\{l_{elastic} = l_{elastic,i} \cdot \left(1 + \varepsilon_x\right)\right\},\right.\right.$
$\left.\left.elastic\_diameter\right)\right);$
$elastic\_diameter\_from\_extension := subs\left(\left\{\varepsilon_x = strain\_clipped, \nu_{elastic} = \nu, d_{elastic,i} = d_e\right\},\right.$
$\left.elastic\_diameter\_from\_extension\right);$
$elastic\_major\_radius := subs\left(\left\{d_{elastic} = elastic\_diameter\_from\_extension\right\}, r_{M,elastic}\left(\varepsilon_x\right)\right);$
$elastic\_major\_radius := subs\left(\left\{d_{elastic,i} = d_e, d_{wire} = d_w, p_i = p, \nu_{elastic} = \nu, \varepsilon_x = strain\_clipped\right\},\right.$
$\left.elastic\_major\_radius\right);$
if $elastic\_major\_radius < 0$ **then**

$cs\_elastic := helix\_cross\_section\left(x, u, \pi, 0, \dfrac{elastic\_diameter\_from\_extension}{2},\right.$

$\left.pitch\_from\_strain(strain\_clipped)\right);$

else

$cs\_elastic := helix\_cross\_section\left(x, u, \pi, r_{M,elastic}(\varepsilon_x), \dfrac{d_{elastic}}{2}, pitch\_from\_strain(\varepsilon_x)\right);$

$cs\_elastic := subs(\{d_{elastic} = elastic\_diameter\_from\_strain(strain\_clipped)\}, cs\_elastic);$

end if;

$cs\_elastic := subs(\{\varepsilon_x = strain\_clipped\}, cs\_elastic);$

$cs\_elastic := subs(\{d_{wire} = d_w, p_i = p, d_{elastic,i} = d_e\}, cs\_elastic);$

$cross\_sections := [cs\_wire\_outer, cs\_wire\_inner, cs\_elastic];$

end proc:

The following function generates a plot of the sensor geometry with overlaid cross-section.

> $PlotSensor :=$ **proc** $(x, d_e, d_w, t_i, p, \nu, n, \varepsilon)$

local $hh, cs, plt\_helix\_wire, plt\_helix\_elastic, plt\_cs\_dielectric, plt\_cs\_wire, plt\_cs\_elastic, plot\_handles;$

$hh := SensorGeometry(d_e, d_w, p, \nu, n, \varepsilon);$

$cs := SensorCrossSection(x, d_e, d_w, t_i, p, \nu, n, \varepsilon);$

$plt\_helix\_wire := plot3d\left(hh[1], t = 0..n \cdot 2 \cdot \pi, u = 0..2 \cdot \pi, color = Color\left(\left[\dfrac{249}{255}, \dfrac{200}{255}, \dfrac{70}{255}\right]\right), transparency = 0.5, scaling = constrained, numpoints = 10000\right);$

$plt\_helix\_elastic := plot3d\left(hh[2], t = 0..n \cdot 2 \cdot \pi, u = 0..2 \cdot \pi, color = Color\left(\left[\dfrac{84}{255}, \dfrac{88}{255}, \dfrac{99}{255}\right]\right), transparency = 0.5, scaling = constrained, numpoints = 10000\right);$

$plt\_cs\_dielectric := spacecurve(cs[1], u = 0..2 \cdot \pi, color = Color([0, 0, 1]), scaling = constrained);$

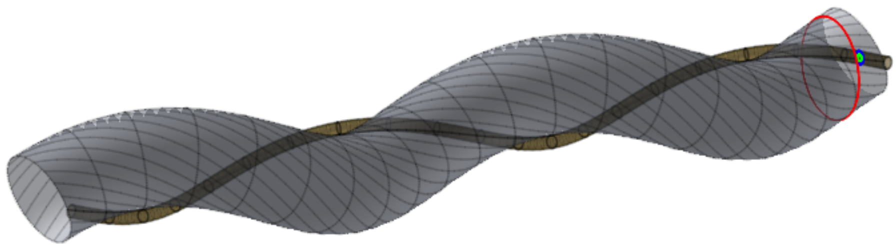
$plt\_cs\_wire := spacecurve(cs[2], u = 0..2 \cdot \pi, color = Color([0, 1, 0]), scaling = constrained);$

$plt\_cs\_elastic := spacecurve(cs[3], u = 0..2 \cdot \pi, color = Color([1, 0, 0]), scaling = constrained);$

$plot\_handles := [plt\_helix\_wire, plt\_helix\_elastic, plt\_cs\_dielectric, plt\_cs\_wire, plt\_cs\_elastic];$

end proc:

$display(PlotSensor(0.5, initial\_elastic\_diameter, wire\_diameter, dielectric\_thickness, initial\_pitch, elastic\_poissons\_ratio, number\_of\_turns, final\_strain));$



> $AnimateSensor := \mathbf{proc}\left(x, d_e, d_w, t_i, p, \nu, n, \varepsilon_{\max}\right)$
local $hh, cs, plt\_helix\_wire, plt\_helix\_elastic, plt\_cs\_dielectric, plt\_cs\_wire, plt\_cs\_elastic, plot\_handles;$

$plt\_helix\_wire := animate\left(plot3d, \left['SensorGeometry'\left(d_e, d_w, p, \nu, n, \varepsilon\right)[1], t = 0..2\cdot\pi, u = 0..2\cdot\pi\right], \varepsilon = 0..\varepsilon_{\max}, frames = 100, color = Color\left(\left[\frac{249}{255}, \frac{200}{255}, \frac{70}{255}\right]\right), transparency = 0.5, scaling = constrained\right):$

$plt\_helix\_elastic := animate\left(plot3d, \left['SensorGeometry'\left(d_e, d_w, p, \nu, n, \varepsilon\right)[2], t = 0..2\cdot\pi, u = 0..2\cdot\pi\right], \varepsilon = 0..\varepsilon_{\max}, frames = 100, color = Color\left(\left[\frac{84}{255}, \frac{88}{255}, \frac{99}{255}\right]\right), transparency = 0.5, scaling = constrained\right):$

$plt\_cs\_dielectric := animate\left(spacecurve, \left['SensorCrossSection'\left(x, d_e, d_w, t_i, p, \nu, n, \varepsilon\right)[1], u = 0..2\cdot\pi\right], \varepsilon = 0..\varepsilon_{\max}, frames = 100, color = Color([0, 0, 1]), transparency = 0.5, scaling = constrained\right):$

$plt\_cs\_wire := animate\left(spacecurve, \left['SensorCrossSection'\left(x, d_e, d_w, t_i, p, \nu, n, \varepsilon\right)[2], u = 0..2\cdot\pi\right], \varepsilon = 0..\varepsilon_{\max}, frames = 100, color = Color([0, 1, 0]), transparency = 0.5, scaling$

$= constrained\big)$:

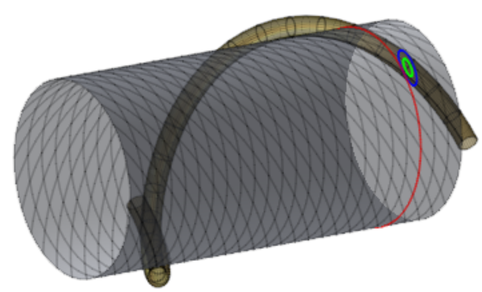
$plt\_cs\_elastic := animate\big(spacecurve, \big['SensorCrossSection'\big(x, d_e, d_w, t_i, p, \nu, n, \varepsilon\big)[3], u = 0 ..2\cdot\pi\big], \varepsilon = 0 ..\varepsilon_{max}, frames = 100, color = Color([1, 0, 0]), transparency = 0.5, scaling = constrained\big)$:

$plot\_handles := [plt\_helix\_wire, plt\_helix\_elastic, plt\_cs\_dielectric, plt\_cs\_wire, plt\_cs\_elastic]$;

end proc:

$display(AnimateSensor(0.5, initial\_elastic\_diameter, wire\_diameter, dielectric\_thickness, initial\_pitch, elastic\_poissons\_ratio, number\_of\_turns, 1), insequence = false)$

$\varepsilon = 0.$



\> $CrossSectionToBitmap :=$ **proc**$\big(x, d_e, d_w, t_i, p, \nu, n, \varepsilon\big)$

local $cs, plt\_cs\_dielectric, plt\_cs\_wire, plt\_cs\_elastic, plot\_handles$;

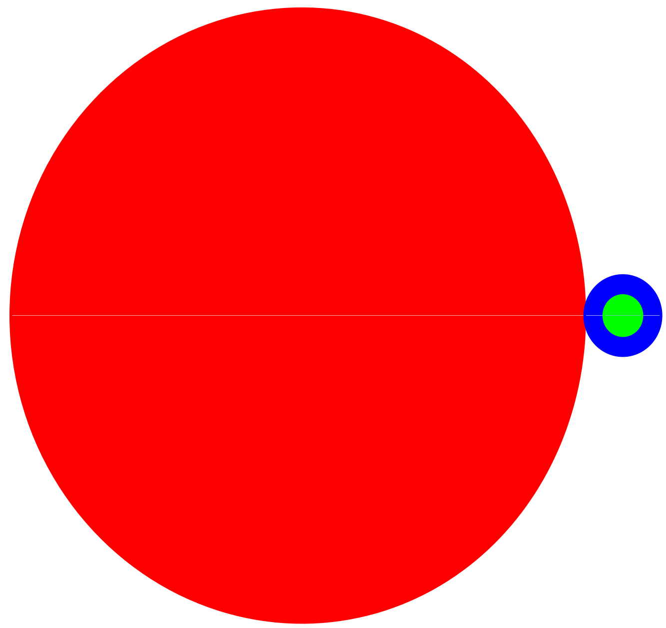
$cs := SensorCrossSection\big(x, d_e, d_w, t_i, p, \nu, n, \varepsilon\big)$;

$plt\_cs\_dielectric := plot\big([cs[1][2], cs[1][3], u = 0 ..2\cdot\pi], scaling = constrained, filled = [color = Color([0, 0, 1]), transparency = 0], color = Color([0, 0, 1]), thickness = 0\big)$;

$plt\_cs\_wire := plot\big([cs[2][2], cs[2][3], u = 0 ..2\cdot\pi], scaling = constrained, filled = [color = Color([0, 1, 0]), transparency = 0], color = Color([0, 1, 0]), thickness = 0\big)$;

$plt\_cs\_elastic := plot\big([cs[3][2], cs[3][3], u = 0 ..2\cdot\pi], scaling = constrained, filled = [color = Color([1, 0, 0]), transparency = 0], color = Color([1, 0, 0]), thickness = 0\big)$;

plot_handles ≔ [*plt_cs_elastic*, *plt_cs_wire*, *plt_cs_dielectric*];
end proc:
display(*CrossSectionToBitmap*(0, *initial_elastic_diameter*, *wire_diameter*, *dielectric_thickness*, *initial_pitch*, *elastic_poissons_ratio*, *number_of_turns*, *final_strain*), *axes* = *None*)



Remember to disable antialiasing in Maple settings!

> *plotsetup*(*bmp*, *plotoutput* = *cat*("mask_", *run_name*, "_p", *convert*(*initial_pitch*, *string*), ".bmp"), *plotoptions* = "width=1600,height=1600") :
display(*CrossSectionToBitmap*(0, *initial_elastic_diameter*, *wire_diameter*, *dielectric_thickness*, *initial_pitch*, *elastic_poissons_ratio*, *number_of_turns*, *final_strain*), *axes* = *None*)